

09871 10000

TO ALL WHOM IT MAY CONCERN:

## WORKFLOW ENCAPSULATION IN STATELESS ENVIRONMENTS

Specification  
Attorney Docket No. 0544MH-40021  
Page 1

## 1

2

## 6

## 7

8

## 11

12

15



## 1

12

05686731 101000

**BRIEF DESCRIPTION OF THE DRAWINGS**

Additional objects, features and advantages will be apparent in the written description which follows.

Figure 1 is a block diagram of a series of interrelated web pages;

Figure 2 is a state diagram of control steps corresponding to the diagram of Figure 1;

Figure 3 is a block diagram illustrating a preferred system architecture in accordance with the present invention; and

Figures 4 and 5 are flow charts showing operation of the system of Figure 3.

1

2

9

16







1 Third, the workflow module is responsible for directing the course of  
2 interactions with the user. After processing a request, the workflow module  
3 determines the appropriate response and causes an appropriate  
4 presentation to be made to the user's interface. The workflow module  
5 generates logical views of the information to be presented to the users,  
6 which is converted to a physical view to be presented to the user.

7 The presentation portion of the application consists of a number of  
8 views, roughly corresponding to web pages in most applications, which  
9 contain the information to be presented to each user. The job of the  
10 workflow module is to identify the next view to be presented, and provide  
11 information which must be used to provide data within that view. The  
12 presentation portion of the application handles the task of formatting the view  
13 appropriately to be presented to the user, and all other details of the user  
14 interface itself. Thus, the presentation of information to the user is separated  
15 from the logical flow of the underlying business process. As described  
16 below, this provides a great flexibility for web-based applications.

17 Referring to Figure 3, a system for executing applications to interface  
18 with remote users is designated generally with reference number 30.  
19 Content engines 32, 34 are connected to interfaces 36, 38 respectively.  
20 Both content engines 32, 34 are connected to a single set of process  
21 modules 40. Each content engine is connected to configuration data 42, 44,  
22 and to a channel adapter 46, 48. Each channel adapter 46, 48 is connected

09686731-101000

1 to a set of views 50, 52 respectively. Views 50, 52 are also connected to  
2 interfaces 36, 38 respectively.

3 Two content engines 32, 34 are illustrated to show the value of the  
4 present approach in dealing with different types of user interfaces. Interface  
5 36 can be, for example, a web based server which communicates with  
6 remote users over the internet in a known fashion. Interface 38 can be a  
7 completely separate type of interface, such as an audio interface intended to  
8 be used over the telephone. Although user interfaces for an internet based  
9 computer and a telephone present completely different interfaces to an end  
10 user, they can both be used to implement the same kind of underlying  
11 business transaction. The present invention allows a single business  
12 transaction to be defined which can be used successfully with radically  
13 different types of interfaces.

14 The content engine 32 functions as a central manager and router for  
15 all requests received from a remote user. Requests are communicated from  
16 remote users to interface 36, which passes them along to content engine 32.  
17 Content engine 32 determines which process module should handle the  
18 request, and routes the request to that process module for processing.  
19 When a response is received from the process module, it is fed back to the  
20 user through channel adapter 46, views 50, and interface 36.

21 The content engine 32 provides various services to the process









1 In the more typical case, where the request is valid, the process  
2 module handles the request. This handling of user requests typically involves  
3 retrieving data from the business logic layer, initiating transactions and  
4 updating the transient state of the system. The process module then  
5 decides what the appropriate response (view) is to show the user based on  
6 the new state of the system. This decision is communicated to the content  
7 engine 32, which performs the actual selection and manages the rendering  
8 of a physical view to be presented back to the user. In rendering the view,  
9 the current state data of the process module is made available to the view  
10 through a channel-independent mechanism.

11 The purpose of the channel adapter 46 is to provide an extensible  
12 mechanism whereby the content engine 32 can manage the presentation of  
13 content developed in multiple authoring environments. The content engine  
14 32 resolves a logical view into a physical view. Based on the content type of  
15 the physical view, the content engine 32 then calls on a specific channel  
16 adapter 46 to resolve that view. It is the responsibility of the channel adapter  
17 46 to provide the state data of the process module to the view in a channel-  
18 specific way and manage the rendering of that view.

19 Channel adapters 46, 48 thus allow views to be developed in any  
20 number of authoring environments. For instance web pages may be  
21 developed using ASP, JSP, XSL, Cold Fusion or other environments. It is  
22 then the responsibility of a channel adapter for that specific authoring





1 as establishing parameters such as the expertise and the identify of the user  
2 which can affect which views are to be presented. Once the new workflow  
3 has been instantiated 66, the incoming request is passed to it 68.

4 If the incoming request is made with respect to an existing workflow  
5 module, that module is restored 70 and the request is passed to it 68.  
6 Between calls to a process module, the state of the module is saved to a  
7 temporary memory, sometimes referred to as "persisting its state". Between  
8 requests, the process module is not doing anything. It is reactivated from  
9 temporary storage only when a request is received, and will be returned to  
10 an inactive state after operations on that request are complete.

11 This restoration allows state information to be retained in what is  
12 essentially a stateless environment. By instigating a new workflow module  
13 for each session, all can operate independently and properly retain state.

14 After the request is passed to the workflow module, various workflow  
15 operations are performed 72. These operations will be detailed further in  
16 connection with Figure 5. After the process module performs its workflow  
17 operation 72, a logical view to be presented to the user is returned 74. Along  
18 with the identification of the logical view is all data which is necessary to be  
19 returned to the user in response to the request just handled. This can be, for  
20 example, information such as confirmation of an order, pricing information



09586731-101000

1 retrieving data from the underlying business system 96, initiating  
2 transactions 98, and updating the underlying system 100, are typical actions  
3 undertaken by process modules. It may not be necessary to perform any or  
4 all of these steps in any particular state; the actual steps to be performed are  
5 application specific and determined by the current state of the process  
6 module and the user input.

7       The processes performed are made with the underlying business  
8 system. For example, goods can be ordered, data bases updated, and data  
9 retrieved to be presented to the user. All of these steps which occur are  
10 transparent to the user, with only the end result being returned. After all  
11 application logic steps 96-100 are performed, the process module  
12 determines the next state into which it should change 102, and returns an  
13 identification of a logical view to the content engine 104. Along with an  
14 identification of this logical view is all information necessary to be placed into  
15 the view for presentation to the user.

16       The above description has been with reference to content engine 32.  
17 The same process modules 40 used with content engine 32 can also be  
18 used with content engine 34 which delivers views into a different channel.  
19 The underlying process modules encapsulate the underlying business  
20 workflow, such as the process of taking and confirming an order. If that  
21 order is taken over a channel such as a telephone, limited to either voice  
22 recognition or entry of data using a telephone key pad, the presentations to



1        While the invention has been shown in only one of its forms, it is not  
2        thus limited but is susceptible to various changes and modifications without  
3        departing from the spirit thereof.

09686731-101000